

To Do List Q (TDLQ) documentation

1. index.php is the file that is loaded first by Apache when the website is visited.

2. The include lines like this:

```
include("config/dbSession.class");
```

Include other files used in the software.

3. dbSession.class is a set of functions for doing generic database reads, writes etc.

4. The most different page is the user modification page as it uses AJAX and not forms. This is the page to be understood the most. That mixed with using PHP classes. W3 Schools has some great tutorials on classes and AJAX.

Trouble shooting

1. Start at index.php

2. To comment out individual lines use // at the beginning of the line.

3. To comment out multiple lines embrace the lines in /** and */.

4. Code arrangement of most files involved is:

TITLE

INCLUDES – Points to other files which are read before continuing.

CONNECT TO DB

SWAP OVER ANY POST OR GET VARIABLES

This is went redirected back to index.php from another page.

GET picks up the vars passed by a hyperlink shown at the end of the URL after ?= eg

Paste example here

POST are vars passed from a html form and are not seen in the URL but hidden in the background.

DEBUG VARIABLES HERE – Where you can echo back variables you've either created or have been passed from a previous page.

CATCHES – If the page is being visited from another page or is reloading itself from a html form then variables will be passed to the page as either POST or GET variables. The CATCHES

section reads the `$_POST['OptionCatch']` variable and if there's any match to the conditional 'case' check then the relevant code runs. The general method here is to point to a function in one of the above includes.

Example:

```
case "reminder_card";  
$ReminderID = $_POST['ReminderID'];  
//echo "\$ReminderID = $ReminderID <BR>";  
reminder_card();          // Found in reminder_functions.php  
break;
```

MAIN is the main area of code. If no catches are found then the MAIN area runs by default usually pointing to a function at the bottom of the page.

FUNCTIONS - All functions are kept at the bottom of the page.

5.

Method I

- a. To see where the code is failing in the first line echo something back to the screen.

Example

```
<?PHP echo "This is FUBAR!" ?>
```

- b. If "This is FUBAR!" it does not appear in the web browser then make a blank file called `test_file.php` and enter the code above as the only line.
- c. If that doesn't work then there's something wrong with the web server.
- d. If it does work then comment out the rest of the code using `/**` and `*/`
- e. If the echo does work then slowly move the `/**` down the code until it fails again.

Method II

- a. To see where the code is failing in the first line echo something back to the screen.

Example

```
<?PHP echo "This is FUBAR!" ?>
```

- b. If it does work then go down several lines in the code and enter

```
<?PHP echo "This is FUBAR!" ?>
```

again. When the web page stops echoing this message back to you, you know the error is somewhere above the last echo.